Maximum Mean Discrepancy for Generalization in the Presence of Distribution and Missingness Shift

Anonymous Author(s)
Affiliation
Address
email

Abstract

Covariate shifts are a common problem in predictive modeling on real-world problems. This paper proposes addressing the covariate shift problem by minimizing Maximum Mean Discrepancy (MMD) statistics between the training and test sets in either feature input space, feature representation space, or both. We designed three techniques that we call MMD Representation, MMD Mask, and MMD Hybrid to deal with the scenarios where only a distribution shift exists, only a missingness shift exists, or both types of shift exist, respectively. We find that integrating an MMD loss component helps models use the best features for generalization and avoid dangerous extrapolation as much as possible for each test sample. Models treated with this MMD approach show better performance, calibration, and extrapolation on the test set.

1 Introduction

Machine learning models often work on the assumption that the training and test datasets follow the same distribution. In practice, this might not be the case, yet the difference in distribution, called dataset shift, is often ignored. This might be acceptable when the dataset shift is small, but small scale shift is not always guaranteed. When it is not, it can lead to poor performance on shifted test sets. Dataset shift can be divided into three main categories: covariate shift, prior probability shift, and concept shift. Here we focus on covariate shifts that occurs when the distribution of input features changes [20].

There have been some attempts to address covariate shift in previous literature. One straightforward approach is to identify the shifted covariates and drop them from the feature set. However, this can be challenging for high-dimensional data, and one may also lose useful information in the process. A softer method is to use importance re-weighting by assigning higher weights to training samples that are more similar to the samples in the test dataset. The importance weights can be calculated through various techniques, including density estimation [21, 14, 29, 5], kernel mean matching (KMM) [11], Kullback-Leibler importance estimation [24], and discriminative learning [3]. These methods work particularly well for sample selection bias, but they treat samples at a row/sample level which is not flexible enough.

In this paper, we propose optimizing an auxiliary maximum mean discrepancy (MMD) loss with a mixture of kernels to treat the covariate shift problem. MMD is already widely used in unsupervised domain adaptation (UDA) [26, 18, 19, 28, 4]. Here we extend it to the more general covariate shift problem. Conceptually, we can think of domain adaptation as an extreme case of covariate shift as the joint distribution of input variables shifts to a different domain entirely.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Moreover, previous UDA works has never considered an important special case of covariate shift that involves data missingness. For example, in an internal project of carbon emission prediction in Section 3, we try to train a model with the companies who reported carbon emission to infer carbon emission for the unreported companies. However, we find that the unreported companies usually have higher missing rates in various sets of features compared to the reported companies. If we simply ignore the data missingness shift, the model performance may be degraded on the unlabeled set. In this paper, we propose building a novel masker model to mask training samples using MMD and thereby align the missingness patterns in the training and test datasets. We find that this can help improve the test performance by preventing the model from depending on relationships that will be unavailable at test time.

Expanding on this, we combine a missingness shift treatment and a general feature distribution shift treatment together in a hybrid model, and show that the hybrid model is superior through various experiments done with both synthetic data and real data.

2 Method

2.1 Preliminary: Maximum Mean Discrepancy

MMD is a non-parametric metric to measure the distance between two distributions. Therefore, to compare two distributions, one does not need to know their probability density functions, or PDFs. This is useful as the PDFs can be difficult to calculate or can even be unknown. Assume we have two sets of samples $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$ drawn from two distributions $P(X)$ and $P(Y)$. Let k be a measurable and bounded kernel of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ of functions, then the empirical estimation of MMD between the two distributions in $\mathcal{H}_k$ can be written as

$$\mathcal{L}_{MMD^2}(X,Y) \ = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(x_i, x_{i'}) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(y_j, y_{j'}) - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(x_i, y_j) \tag{1}$$

When the underlying kernel is characteristic [7, 23], MMD is zero if and only if $P(X) = P(Y)$ [9]. For example, the popular Gaussian RBF kernel, $k(x, x') = \exp(-\frac{1}{2\sigma^2}|x - x'|^2)$, is a characteristic kernel and was widely used in previous literature [17, 25, 19, 16]. The bandwidth parameter σ in the Gaussian RBF kernel can affect the hypothesis testing power of MMD measurement. There were some heuristics in previous literature [22, 10], but later research [17, 25, 16] found that using a mixture of 5 or more bandwidths performed well, so we followed this procedure as well.

2.2 MMD Representation, MMD Mask, MMD Hybrid

Let us first define the context of our predictive problem. Assume that we have training and test features $X_{tr}, X_{te} \in \mathcal{R}^m$, where m is the number of features. We also know the target value y_{tr} for the training set, and we want to build a model to predict y_{te} for the test set. When we train the model with (X_{tr}, y_{tr}), a covariate shift between X_{tr} and X_{te} can cause the trained model to perform poorly on the test set.

Assume we build a neural network model for the task, then similar to what has been widely used in domain adaptation, we can match the MMD statistics between emb_{tr} and emb_{te} at some intermediate embedding layer so that the learned embeddings have similar distributions between the training and test sets and will therefore reduce the amount of extrapolation. We call this approach **MMD Representation**. Note that models like DAN [18] and JAN [19] can be considered as special forms of MMD Representation, as they also apply MMD to multiple specific intermediate layers. The final loss we will optimize is the original task loss plus the MMD loss in the representation space:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda \mathcal{L}_{MMD^2}(emb_{tr}, emb_{te}) \tag{2}$$

where λ is a hyperparameter to adjust the ratio between the original loss and the MMD loss. Generally, we found that setting λ to match the magnitudes of the two loss terms leads to the best results. Com-

pared to the approach of dropping features with covariate shift before training, MMD Representation is able to retain as much information as possible from each individual feature and find a common feature space that has as little shift as possible between the training and test sets.

Despite the advantages of MMD Representation, we found that it is insufficient when a missingness shift is involved, as was the case for the carbon estimation problem mentioned in Section 1. Denote I_{tr} and I_{te} as the original missingness indicators for the training and test features X_{tr} and X_{te}, respectively; we want to learn a new missingness indicator $\hat{I}_{tr}$ conditioned on X_{tr} and I_{tr} that can minimize the MMD loss between the joint distributions $P(X'_{tr}, \hat{I}_{tr})$ and $P(X_{te}, I_{te})$:

$$\mathcal{L} = \mathcal{L}_{MMD^2}((X'_{tr}, \hat{I}_{tr}), (X_{te}, I_{te})) \tag{3}$$

where X'_{tr} is generated by masking the original X_{tr} using learned $\hat{I}_{tr}$. The downstream model is then trained using X'_{tr} instead of X_{tr}. In this way, the downstream model can focus on the right features for each training sample. Importance resampling using kernel mean matching also operates in the original feature space, but it removes or downweights each sample as a whole (at the row level), while we treat each sample feature by feature.

Furthermore, we can combine MMD Representation and MMD Mask together as a more comprehensive shift treatment. We call this **MMD Hybrid**. Denote the masker model as f_M, the feature extractor as f_F, and the prediction network as f_P. We use an alternating update method to train the MMD Hybrid model, see Algorithm 1 in Appendix for more details. See Figure 3 in Appendix for the architectural details for all three MMD methods.

3 Evaluation

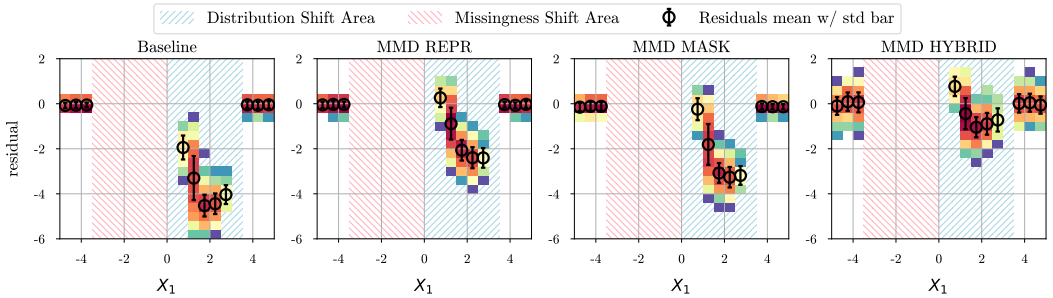
Due to the space limit, we leave all the data and experiments details in Appendix C. We first used synthetic data to *gain a general understanding of which MMD models work best in which situations*. The synthetic data has two features X_1 and X_2, both having some predictive power for the target y. Although X_1 can predict y more accurately compared to X_2, we designed a distribution shift in the positive range of X_1, and a missingness shift in the negative range of X_1 between the training and test datasets. See Appendix C.1 for more details on how to generate the synthetic data.

Figure 1 shows how each model performs in different ranges of X_1 and X_2 on the test data. In the region where data shift does not exist, all models learn the relationship very well and have very small residuals. In the region where missingness shift exists, the MMD Representation reduced residuals significantly compared to the baseline model, but not as well as the MMD Mask and the MMD Hybrid. In the region where distribution shift exists, we see that the baseline model is worst, followed by MMD Mask, MMD Representation and MMD Hybrid, in order.

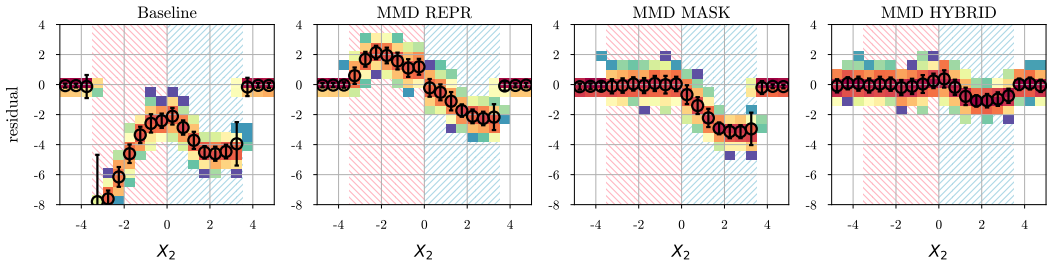
We further tested *how our approaches compare to exisitng methods generally*. Table 1 shows the performance metrics for each model from 10 runs on the test set for our synthetic data, a Bike Sharing dataset [1] and an IEEE-CIS Fraud Detection dataset [2]. Our MMD Hybrid almost always performs best in all three datasets.

Table 1: Averages and standard deviations of performance metrics (MSE for synthetic data, RMSE for the Bike Sharing and AUC for the IEEE-CIS Fraud Detection) of 10 runs on test data for each model.

MODEL	SYNTHETIC DATA	BIKE SHARING	FRAUD DETECTION
BASELINE	17.682 ± 9.911	116.2 ± 3.9	$86.29\% \pm 0.59\%$
KMM[11]	17.666 ± 9.854	116.5 ± 5.2	N.A.
DAN[18]	0.753 ± 0.698	106.3 ± 2.3	$85.91\% \pm 0.79\%$
JAN[19]	0.820 ± 0.693	107.4 ± 2.2	$86.30\% \pm 0.54\%$
MMD REPR	2.303 ± 1.373	107.7 ± 1.7	$86.44\% \pm 0.48\%$
MMD MASK	2.573 ± 1.119	106.1 ± 7.0	$\mathbf{87.25\% \pm 0.19\%}$
MMD HYBRID	$\mathbf{0.331 \pm 0.201}$	$\mathbf{98.7 \pm 3.8}$	$87.22\% \pm 0.32\%$



(a) residuals v.s. X_1



(b) residuals v.s. X_2

Figure 1: Synthetic data: residuals v.s. feature X_1 and X_2 on test set from the baseline model, MMD Representation model, MMD Mask model and MMD Hybrid model. The heatmap shows the sample frequency in each small cell. The black circles and bars represent the average and standard deviation of residuals in each bucket of X_1/X_2. Distribution shift of X_1 occurs in blue shaded area and missingness shift of X_1 occurs in pink shaded area.

Moreover, we applied MMD Mask in an internal carbon emission estimation project where there is a missingness shift between labeled and unlabeled data. The companies that reported carbon emission (labeled data) usually have more complete feature information, while the unreported companies (unlabeled data) have many missing values in different sets of features. In this case, *the predictive model is a tree-based model without representation learning ability, so only MMD Mask can be applied here*. As a baseline to compare with our MMD Mask model, we also tried a Simple Model that randomly sampled masks from unlabeled data conditioned on industry.

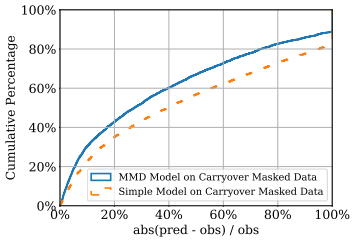


Figure 2: Carbon emission estimation: cumulative percentages for the percentage error of prediction. MMD Model v.s. Simple Model performance on mask-carryover set.

Figure 2 shows the percentage error of prediction from the MMD Model and the Simple Model on the masked dataset by carrying over masks for companies from a year in which they did not report carbon emissions to a following year in which they did. Note that this is the most realistic set of masks we can get for the labeled data. As we can see, the MMD Model has a large performance gain over the Simple Model.

To conclude, we proposed a novel MMD masker model to match the joint distribution of input features and missingness indicators in input space between the training and test datasets, an approach designed specifically for treating a missingness shift, which has never been considered before. We have shown that MMD Representation works better for distribution shifts while MMD Mask works better for missingness shifts on both synthetic data and real data. Another advantage of MMD Mask is that it can be applied even when the downstream model cannot learn an intermediate representation such as in a tree-based model. Furthermore, we combined MMD matching in the feature embedding space and in the raw input space to yield superior results.

References

[1] https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset.

[2] https://www.kaggle.com/c/ieee-fraud-detection/overview.

[3] Bickel, K., M. Brückner, and T. Scheffer (2009). Discriminative learning under covariate shift. *Journal of Machine Learning Research 10*, 2137–2155.

[4] Chen, C., Z. Fu, S. Jin, Z. Cheng, X. Jin, and X. Hua (2020). Homm: Higher-order moment matching for unsupervised domain adaptation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, Palo Alto, CA, USA, pp. 3422–3429. AAAI Press.

[5] Dudík, M., R. E. Schapire, and S. Phillips (2005). Correcting sample selection bias in maximum entropy density estimation. *Advances in Neural Information Processing Systems 18*, 323–330.

[6] Fanaee-T, H. and J. Gama (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 1–15.

[7] Fukumizu, K., A. Gretton, X. Sun, and B. Schölkopf (2008). Kernel measures of conditional dependence. In B. H. Schölkopf, J. C. Platt, and T. Hoffman (Eds.), *Proceedings of the 20th Neural Information Processing Systems (NIPS 2007)*, Red Hook, NY, USA, pp. 489–496. Curran Associates Inc.

[8] Gretton, A., K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola (2007). A kernel method for the two-sample-problem. In B. H. Schölkopf, J. C. Platt, and T. Hoffman (Eds.), *Proceedings of the 19th Neural Information Processing Systems (NIPS 2006)*, Cambridge, MA, USA, pp. 513–520. MIT Press.

[9] Gretton, A., K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola (2012). A kernel two-sample test. *Journal of Machine Learning Research 13*, 723–773.

[10] Gretton, A., D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur (2012). Optimal kernel choice for large-scale two-sample tests. *Advances in Neural Information Processing Systems*, 1205–1213.

[11] Huang, J., A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf (2007). Correcting sample selection bias by unlabeled data. In B. H. Schölkopf, J. C. Platt, and T. Hoffman (Eds.), *Proceedings of the 19th Neural Information Processing Systems (NIPS 2006)*, Cambridge, MA, USA, pp. 601–608. MIT Press.

[12] Jang, E., S. Gu, and B. Poole (2017). Categorical reparameterization with gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

[13] Jitkrittum, W., W. Xu, Z. Szabó, K. Fukumizu, and A. Gretton (2018). A linear-time kernel goodness-of-fit test. In U. von Luxburg, I. Guyon, S. Bengio, H. Wallach, and R. Fergus (Eds.), *Proceedings of the 31st Neural Information Processing Systems (NIPS 2017)*, Red Hook, NY, USA, pp. 261–270. Curran Associates Inc.

[14] Kanamori, T. and H. Shimodaira (2000). Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference 116*, 149–162.

[15] LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*(11), 2278–2324.

[16] Li, C., W. Chang, Y. Cheng, Y. Yang, and B. Póczos (2018). Mmd gan: Towards deeper understanding of moment matching network. In U. von Luxburg, I. Guyon, S. Bengio, H. Wallach, and R. Fergus (Eds.), *Proceedings of the 31st Neural Information Processing Systems (NIPS 2017)*, Red Hook, NY, USA, pp. 2200–2210. Curran Associates Inc.

[17] Li, Y., K. Swersky, and R. Zemel (2015). Generative moment matching networks. In F. Bach and D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, Lille, France, pp. 1718–1727. JMLR.org.

[18] Long, M., Y. Cao, J. Wang, and M. I. Jordan (2015). Learning transferable features with deep adaptation networks. In F. Bach and D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, Lille, France, pp. 97–105. JMLR.org.

[19] Long, M., H. Zhu, J. Wang, and M. I. Jordan (2017). Deep transfer learning with joint adaptation networks. In D. Precup and Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, Sydney, Australia, pp. 2208–2217. PMLR.

[20] Quinonero-Candela, J., M. Sugiyama, A. Schwaighofer, and N. D. Lawrence (Eds.) (2008). *Dataset Shift in Machine Learning*. Cambridge, MA, USA and London, England: The MIT Press.

[21] Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference 90*, 227–244.

[22] Sriperumbudur, B. K., K. Fukumizu, A. Gretton, and G. R. Lanckriet (2010). Kernel choice and classifiability for rkhs embeddings of probability distributions. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (Eds.), *Proceedings of the 22nd Neural Information Processing Systems (NIPS 2009)*, Red Hook, NY, USA, pp. 1750–1758. Curran Associates Inc.

[23] Sriperumbudur, B. K., A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet (2010). Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research 11*, 1517–1561.

[24] Sugiyama, M., S. Nakajima, H. Kashima, P. V. Bünau, and M. Kawanabe (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. In B. H. Schölkopf, J. C. Platt, and T. Hoffman (Eds.), *Proceedings of the 20th Neural Information Processing Systems (NIPS 2007)*, Red Hook, NY, USA, pp. 1433–1440. Curran Associates Inc.

[25] Sutherland, D. J., H. Tung, H. Strathmann, S. De, A. Ramdas, A. J. Smola, and A. Gretton (2017). Generative models and model criticism via optimized maximum mean discrepancy. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

[26] Tzeng, E., J. Hoffman, N. Zhang, K. Saenko, and T. Darrell (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

[27] van der Maaten, L. and G. Hinton (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research 9*, 2579–2605.

[28] Yan, H., Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo (2017). Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Los Alamitos, CA, USA, pp. 2272–2281. IEEE Computer Society.

[29] Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In R. Greiner and D. Schuurmans (Eds.), *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, New York, NY, USA, pp. 114–122. Association for Computing Machinery.

[30] Zhao, S., J. Song, and S. Ermon (2019). Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, Palo Alto, CA, USA, pp. 5885–5892. AAAI Press.

[31] Zhong, Z., L. Zheng, G. Kang, S. Li, and Y. Yang (2020). Random erasing data augmentation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, Palo Alto, CA, USA, pp. 13001–13008. AAAI Press.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix
   (c) Did you discuss any potential negative societal impacts of your work? [No] There is no negative societal impacts
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix, but note carbon data and corresponding codes are proprietary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 1 and Table 1
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Related Work

MMD was first introduced in the two-sample tests [8]. It is a non-parametric metric to measure the distance between two distributions. Therefore, to compare two distributions, one does not need to know their probability density functions, or PDFs. This is useful as the PDFs can be difficult to calculate or can even be unknown. Because of its simplicity, MMD has been widely applied to various problems, such as goodness-of-fit testing [13], MMD GAN [17, 25, 16], and MMD VAE [30].

One technique related to using MMD for covariate shift is using kernel mean matching to re-weight training samples [11]. In that paper, the authors formulated a quadratic problem between empirical means to find suitable weights for training samples, after which they could apply weighted ordinary least squares or support vector machines. The difference is that they tackled the problem from the importance resampling angle and used the results for more traditional machine learning methods. We incorporate MMD directly into a neural network model in one step instead of two by using the representation learning ability of neural networks, and are able to treat covariate shift in a more granular level than the row/sample level if necessary.

The most related application of MMD to our work is in UDA. Deep Domain Confusion (DDC) [26] is one of the earliest attempts to use MMD for domain invariant learning. Then, Deep Adaptation Networks (DAN) [18] were proposed to match an optimal multi-kernel MMD to reduce domain discrepancy. This was followed by Joint Adaptation Networks (JAN) [19], which were proposed to apply joint MMD in multiple domain-specific layers across domains. Yan et al [2017] proposed adjusting plain MMD by class-specific auxiliary weights to account for the class weight bias across domains. Chen et al [2020] proposed matching higher-order moments to perform fine-grained domain alignment. Note that these applications all focused on domain shifts in classification problems, while we extend the application to more general distribution shift and to missingness shift that has never been considered before.

B Model Details

We show model architectures for MMD Representation, MMD Mask and MMD Hybrid all in Figure /reffig:architecture. Algorithm for MMD Hybrid is detailed in Algorithm 1.

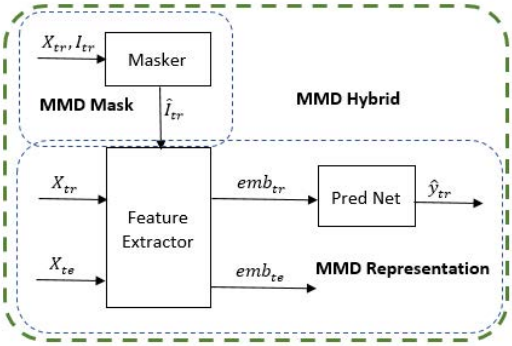


Figure 3: Architecture for MMD Representation, MMD Mask and MMD Hybrid. The two blue dashed boxes contain the architectures for MMD Mask and MMD Representation respectively, and the green dashed box contains the architecture for MMD Hybrid, which combines MMD Mask and MMD Representation.

Algorithm 1 MMD Hybrid

Input: X_{tr}, I_{tr}, X_{te}, I_{te}, y_{tr}, λ

Initialize f_M, f_F, f_P

for each epoch **do**
  for each batch **do**
    $\hat{I}_{tr} \leftarrow f_M(X_{tr}, I_{tr})$
    $X'_{tr} \leftarrow$ Mask X_{tr} by $\hat{I}_{tr}$
    Update $f_M \leftarrow$
      $\min \mathcal{L}_{MMD^2}((X_{tr}, \hat{I}_{tr}), (X_{te}, I_{te}))$

    $emb_{tr}, emb_{te} \leftarrow f_F(X'_{tr}), f_F(X_{te})$
    $\hat{y}_{tr} \leftarrow f_P(emb_{tr})$
    Update f_F, $f_P \leftarrow$
      $\min \mathcal{L}_{task}(\hat{y}_{tr}) + \lambda \mathcal{L}_{MMD^2}(emb_{tr}, emb_{te})$
  end for
end for

C Experiments Details

We tested our approaches on both synthetic data and real data. The real data is comprised of a Bike Sharing dataset from the UCI Machine Learning Repository, an IEEE-CIS Fraud Detection dataset from Kaggle, an internal project of carbon estimation, and the image dataset MNIST [15].

For all experiments except carbon estimation and MNIST, the baseline model is a multilayer perceptron (MLP) model. MMD Representation always shares the same architecture, but also matches MMD statistics at the last hidden layer. MMD Mask outputs masks for each feature per sample; we then mask out training samples based on the learned masks, and use that to train the same downstream baseline model. MMD Hybrid combines MMD Representation and MMD Mask. For synthetic data, the Bike Sharing data and the IEEE-CIS Fraud Detection data, the baseline models take features and missing indicators as inputs, and the missing values were imputed using the means from the training set. We didn't impute missing values for carbon estimation because it used a tree-based model that can handle missing values inherently. For all datasets, we use MMD statistics under a mixture of Gaussian RBF kernels with bandwidths of $\{1, 2, 4, 8, 16, 32\}$. The MMD Mask model uses a Relaxed Bernoulli distribution parametrized by a temperature τ to generate a mask probability for each feature. The Relaxed Bernoulli is a continuous approximation to a Bernoulli distribution over $(0, 1)$, similar to the Gumbel-Softmax trick [12] for categorical distributions. In all experiments, we set τ to 0.1 at the start of training and then gradually decrease it to 0.01. We implemented all models in PyTorch. Neural network models were all trained on one single Nvidia GPU with driver version 440.82 and CUDA version 10.2.

For synthetic data, the baseline model is an MLP with 3 hidden layers with a hidden size of 16 for each. The MMD Mask model has 3 hidden layers with hidden sizes of $\{32, 32, 20\}$. For the MMD Representation and MMD Hybrid models, we also tested different hyperparameter values $\lambda \in \{1, 5, 10\}$ and picked the model that performed best. In all models, we used ReLU as our activation function and optimized network parameters using RMSProp with a learning rate of 0.01. All models were trained for 5,000 epochs with full batch.

For the Bike Sharing, the baseline model is an MLP model with 4 hidden layers with hidden size of $\{64, 64, 64, 64\}$, and the MMD Mask model has 3 hidden layers with hidden sizes of $\{512, 512, 128\}$. For the IEEE-CIS Fraud Detection, the baseline model is an MLP model with 4 hidden layers with hidden size of $\{1024, 512, 512, 256\}$, and the MMD Mask model has 3 hidden layers with hidden sizes of $\{512, 512, 256\}$. For both datasets, we used ReLU as our activation function and optimized network parameters using RMSProp. For the MMD Representation and MMD Hybrid models, we also tested different hyperparameter values $\lambda \in \{1, 10, 100\}$. The labeled data is split into 3:1 as train v.s. validation. The hyperparameters are tuned based on the performance on the validation set.

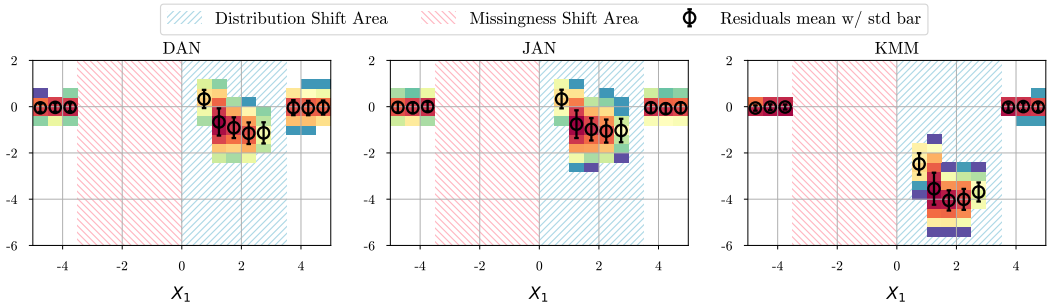
For carbon emission estimtion, the MMD Mask model is an MLP model with 4 hidden layers with hidden size of $\{512, 512, 512, 512\}$, with ReLU activation. We trained the model for 300 epochs with a batch size of 5000, using RMSProp with a learning rate of 0.001.

For all experiments, whenever possible, we also compared our methods with kernel mean matching (KMM) [11], and UDA baselines like DAN [18] and JAN [19]. As for KMM, we followed the procedure in their Empirical KMM optimization Section, and found the best weights for training samples by solving the quadratic problem using cvxopt package[1], then used these weights to reweight samples in the original loss function. Whenever DAN and JAN are used, we apply MMD or Joint MMD to the last two hidden layers as MLPs in this paper are only three to four hidden layers.

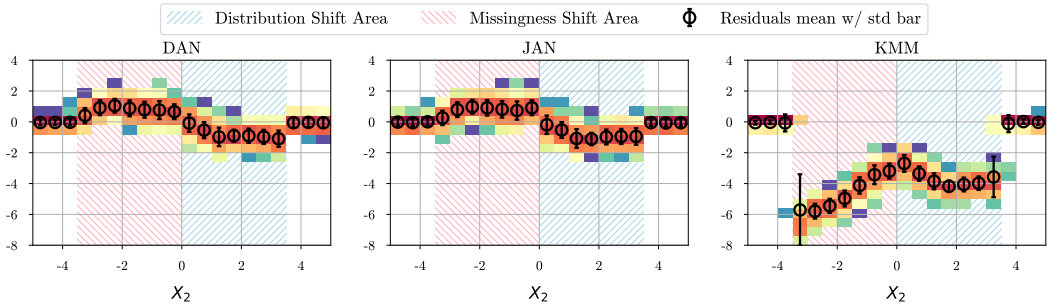
C.1 Synthetic Data

In this experiment, we used synthetic data to gain a general understanding of which MMD models work best in which situations. We constructed a synthetic dataset as shown in Figure 4. There are two input features X_1 and X_2, both having some predictive power for the target y. The relationship

[1] https://cvxopt.org.



(a) residuals v.s. X_1



(b) residuals v.s. X_2

Figure 5: Synthetic data: residuals v.s. feature X_1 and X_2 on test set from the KMM, DAN and JAN. The heatmap shows the sample frequency in each small cell. The black circles and bars represent the average and standard deviation of residuals in each bucket of X_1/X_2. Distribution shift of X_1 occurs in blue shaded area and missingness shift of X_1 occurs in pink shaded area.

between feature X_1 and y is linear while the relationship between X_2 and y is parabolic. We injected noise $\epsilon_1 \sim \mathcal{N}(0, 0.1)$ into feature X_1 and larger-scale noise $\epsilon_2 \sim \mathcal{N}(0, 0.5)$ into feature X_2. Based on this alone, the model should prefer using feature X_1 to feature X_2. However, we also engineered two types of data shifts into feature X_1 between the training and test datasets. We first resampled from the training dataset with replacement to get our test dataset such that both the training and test datasets had 5,000 rows. Then in the test dataset, we designed a missingness shift for X_1 in the range $[-3.5, 0]$ by masking it out, and a distribution shift for X_1 in the range $(0, 3.5]$ by subtracting it by $\epsilon \sim \mathcal{N}(1, 0.1)$ - see Figure 4. Therefore, the test dataset has both missingness and distribution shifts compared to the training dataset. We hope that a properly designed model can learn to predict using feature X_2 instead of X_1 in the regions of X_1 affected by the shifts.

In main sections, we took a scrutiny of MMD Representation, MMD Mask and MMD Hybrid in regions with different types of shift. Here we also include similar analysis for KMM, DAN and JAN. From Figure 5, we can see that KMM, DAN and JAN also have very small residuals in the region where data shift does not exist. When missingness shift or distribution shift exists, performance of all methods deteriorates to different degrees. KMM has a similar performance compared to the Baseline model in both missingness shift and distribution shift regions. In the region where distribution shift exists, DAN and JAN work better than MMD Mask, which is expected. DAN and JAN also work better than MMD Rep-

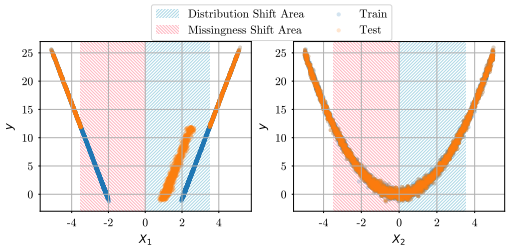
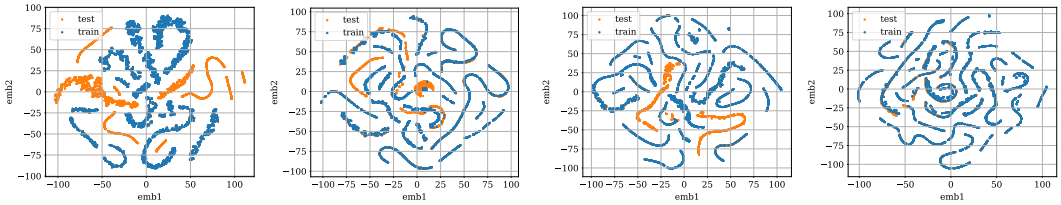


Figure 4: Synthetic data: plots of target y v.s. feature X_1 and X_2 respectively. Distribution shift of X_1 occurs in blue shaded area and missingness shift of X_1 occurs in pink shaded area.



(a) Baseline (b) MMD Representation (c) MMD Mask (d) MMD Hybrid

Figure 6: tSNE plot of embeddings from last hidden layer from each model on synthetic data.

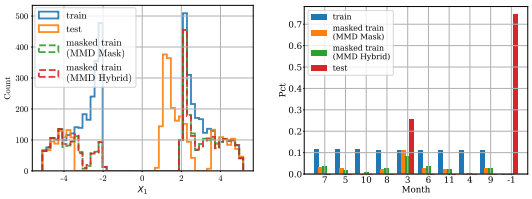
resentation as they try to match more intermediate layers, and work similarly with MMD Hybrid. However, in the region where missingness shift exists, DAN and JAN are not as good as MMD Mask and MMD Hybrid, but similar to MMD Representation.

Furthermore, we also checked the embeddings in the last hidden layer from the training and test phases for each model. Figure 6 shows 2-D tSNE [27] plots of embeddings from the baseline model, MMD Representation, MMD Mask and MMD Hybrid. As expected, in all MMD approaches, the learned embeddings between the training and test sets are more similar to each other than in the baseline model. Among all approaches, MMD Hybrid learned the most consistent embeddings between the training and test sets.

Finally, we examined the masks generated by MMD Mask and MMD Hybrid. Figure 7a shows histograms of features X_1 in the original training set, the test set, and the masked training set. We focused on X_1 here because there is no shift in X_2. As we can see, the masker in MMD Mask and MMD Hybrid learned to mask out the shifted range on both the positive and negative sides to some degree. Since the negative side has purely a missingness shift, it learned better by masking out more samples for $X_1 \in [-3.5, 0]$, while it masked out less samples for $X_1 \in (0, 3.5]$. The masking ability is consistent with the downstream performance we observed.

C.2 Bike Sharing and IEEE-CIS Fraud Detection

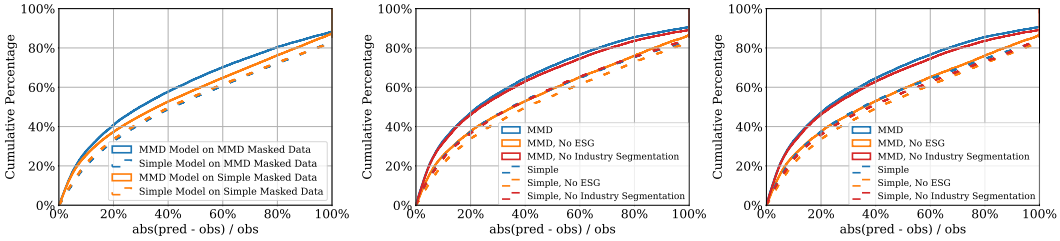
We use the Bike Sharing as the regression example and the IEEE-CIS Fraud Detection as the classification example to show how our MMD models perform in general. The Bike Sharing dataset [6] from UCI contains the hourly and daily bike rental counts in the Capital bikeshare system in Washington, D.C., USA for 2011 and 2012. We focused on hourly data as it provides more samples. The dataset contains features like hour, month, season, workday, holiday and some weather information. We log-transformed the target variable (bike rental counts) due to their long tail, and deliberately selected 2011-03 to 2011-11 as the training data (6,567 rows) and 2011-12 to 2012-03 as the test data (2,917 rows) so that the features have shifts between the training and test data due to the different times of year. Also, since there are some new months in the test set (December, January and February), after ordinal feature processing, they are treated as missing in the prediction stage, which can be viewed as a source of missingness shift.



(a) synthetic data (b) bike sharing data

Figure 7: Comparison of masked training data generated by MMD Mask and MMD Hybrid with original training and test data. (a) Synthetic data: histograms for feature X_1 in original train, test and masked train. (b) The Bike Sharing data: frequency percentage of each month in original train, test and masked training data. -1 represents the new months appearing in test set.

The IEEE-CIS Fraud Detection dataset is from Vesta's real-world-e-commerce transactions and contains a wide range of features from device type to product features. The goal is to estimate fraud probability of the unlabeled test set. We followed the champion model from Chris Deotte and



(a) MMD/simple masked (b) unmasked data (c) masked data

Figure 8: arbon emission estimation: cumulative percentages for the percentage error of prediction. (a) MMD Model / Simple Model performance on MMD Mask/simple mask masked data. (b) MMD Model / Simple Model performance on unmasked data when a set of feature is completely missing. (c) MMD Model / Simple Model performance on masked data when a set of feature is completely missing.

Konstantin Yakovlev[2], and used all the features except 219 Vesta engineered rich features that were determined redundant by their correlation analysis, but did not conduct further feature engineering like they did, as our goal is not to outperform the best model but use this dataset to show how our proposed MMD models can handle distribution and missingness shift in the features and outperform the baseline models. In the end, we used 212 features before one hot encoding the categorical features and adding missing indicators. Missingness shift is observed in some of the 212 features to some degree, but not severe.

We also examined if the masks learned for the month feature in the Bike Sharing dataset meet our expectation. As we can see from Figure 7b, the maskers learn to leave more of March 2011 data in while masking out data from other months more, which matches with our expectation, as March is the only common month in both training and test data. Note that the maskers try to match the joint distribution of all features, not just the marginal distribution of months.

C.3 Carbon Emission Estimation

Carbon Emission Estimation is an internal project that seeks to estimate companies' carbon emissions using various sets of features including environmental, social and governance (ESG) data, industry segmentation data, fundamental financial data, and country and regional data. In total, there are about 1,000 features. One challenge of the project is the missingness shift between labeled and unlabeled data. The labeled data usually has more complete feature information, while unlabeled data has many missing values in different sets of features. As a result, a model trained on the labeled data would generate poor results on the unlabeled data due to the much larger missing rates in various features. We could not take the approach of dropping all features that are sometimes missing as nearly all features in this dataset can be missing. To overcome this challenge, we propose using data augmentation where we generate masks for the labeled data to mimic the missingness patterns in unlabeled data.

In this case, the downstream predictive model is a tree-based model without representation learning ability, so only MMD Mask can be applied here. We trained a masker model to generate masks for labeled data based on all available features by matching the MMD statistics between the labeled and unlabeled data. As a baseline, we also tried using a simple mask model that randomly sampled masks from unlabeled data conditioned on industry. The downstream carbon estimation model was then trained on the masked labeled data with 10 masks sampled from the mask model plus the original training data. We denote MMD Model as the downstream model trained on data masked by masks from the MMD Mask model, and Simple Model as the downstream model trained on data masked by masks from the simple mask model.

To evaluate the results, we designed a series of tests. Besides the one mentioned in the main text, we also tested the performance of the MMD Model and Simple Model on data masked by both MMD

[2]https://www.kaggle.com/cdeotte/xgb-fraud-with-magic-0-9600.

masks and simple masks. We sampled 10 masks from the corresponding mask model and applied those to the original labeled data, then compared predictions from the downstream model trained with two different versions of masked data. Figure 8a compares the percentage error of prediction for each combination. As we can see, the MMD Model always beats the Simple Model whether the data is masked by MMD mask or the simple mask. And when the original labeled data is masked by MMD mask, the performance gain is even larger. The fact that the MMD Model performs better than the Simple Model even on the data masked by the simple mask is interesting. It indicates that the missingness pattern from the simple mask is in a sense contained in the missingness pattern from MMD mask.

In addition, we tested the MMD Model and Simple Model in the specific scenarios that an entire set of features is missing, as this can be the case for certain sub-universes of companies. Figure 8b and Figure 8c plots the scenarios on unmasked data and masked data respectively. Since ESG data is much more useful for estimating carbon emissions than industry segmentation data is, the performance decreases more when ESG data is missing. However, no matter which set of features is entirely missing, the MMD Model always performs better than the Simple Model. Also, if we compare the curves for unmasked data with those for masked data, we can see that most of them are very similar except for the Simple Model with Segmentation data missing. The Simple Model performance on masked data is a bit worse compared to unmasked data when Segmentation data is completely missing. However, MMD Model performance on masked data remains similar to that on unmasked data, indicating that MMD masks are more aligned with the scenarios when an entire set of features is missing.

D MNIST Experiments

In this section, we investigate how our MMD methods perform on more unstructured image data. Our proposed methods can be useful when occlusion is correlated with features/labels on image data. For example, say you have intact images of groundhogs, squirrels and pangolins with labels in training set. But in the unlabeled set, the images of groundhogs and pangolins are incomplete as those are taken when the animals are partially in ground. So we need to learn appropriated masks to mask out right images in training set in order to learn a more accurate and robust classification model.

Table 2: The model architecture for the baseline model in MNIST experiments. It outputs the logits of digits class.

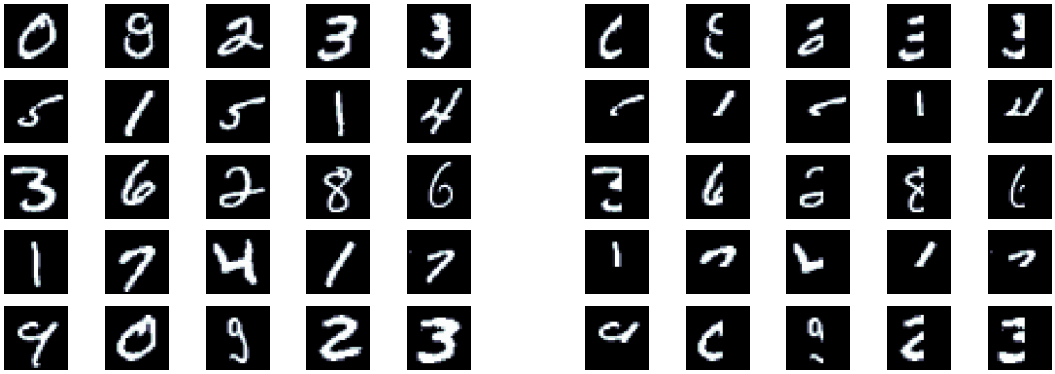
IMAGE $x \in \mathcal{R}^{M \times M \times 1}$
$3 \times 3, stride = 1$ CONV 16 RELU
2×2, MAXPOOL
$3 \times 3, stride = 1$ CONV 32 RELU
2×2, MAXPOOL
DENSE $\rightarrow 10$

Table 3: The model architecture for the MMD Mask in MNIST experiments. It outputs the logits of mask for each pixel.

IMAGE $x \in \mathcal{R}^{M \times M \times 1}$
$3 \times 3, stride = 1$ CONV 16 RELU
$3 \times 3, stride = 2$ CONV 16 RELU
$3 \times 3, stride = 1$ CONV 32 RELU
$3 \times 3, stride = 2$ CONV 32 RELU
DENSE
$4 \times 4, stride = 2$ DECONV 32 RELU
$3 \times 3, stride = 1$ DECONV 32 RELU
$4 \times 4, stride = 2$ DECONV 16 RELU
$3 \times 3, stride = 1$ DECONV 16 LINEAR $\rightarrow 784$

Table 4: Mean and standard deviation of test accuracy from 10 runs for MNIST.

MODEL	ACC MEAN ± STD
BASELINE	0.873 ± 0.030
RANDOM ERASING - V1	0.934 ± 0.008
RANDOM ERASING - V2	0.950 ± 0.004
MMD REPR	0.959 ± 0.005
MMD MASK	0.972 ± 0.002
MMD HYBRID	0.972 ± 0.001
GOLDEN MODEL	0.987 ± 0.001



(a) Original training samples (b) Masked training samples

Figure 9: Original training samples v.s. the training samples masked by the MMD Hybrid model.

To demonstrate our approaches also work on high dimensional image data, we used MNIST which is a handwritten digits dataset where each image is 28×28 pixels. It has 60,000 training images and 10,000 test images. To demonstrate that our methods work when there is a shift, particularly a missingness shift, we created a patterned mask on the original test data. That is, for digits in $\{1, 4, 5, 7\}$, we mask out the last 12 rows of pixels, and for digits in $\{0, 2, 3, 6, 8, 9\}$, we mask out the last 12 columns of pixels.

We compared MMD Representation, MMD Mask, and MMD Hybrid approaches with a baseline model trained on raw training data, a golden model trained with the training samples masked by true masks, as well as traditional data augmentation. All models in this section adopt CNN architectures. The baseline model in MNIST experiments uses the architecture shown in Table 2. The golden model and two variants of data augmentation models also use the exactly same architecture. The MMD Representation uses the same architecture except that it also matches the embeddings between training and test set before entering the final dense layer. The MMD Mask model generates masks using a conv-deconv architecture shown in Table 3. It generates masks to mask original training data, then the masked training data goes through the same downstream model. The MMD Hybrid model combines MMD Representation and MMD Mask. The missing values were imputed as zero (black pixel).

For traditional data augmentation, we tested two versions of RandomErasing [31], with one version always erasing the same area (12×28) of images at a random place and the other erasing a random area ranging from 0 to the same area (12×28) at a random place. We followed the standard procedure to split the training samples into a training set of 54,000 rows and a validation set of 6,000 rows, and trained all models for 20 epochs with a batch size of 64 (except for the MMD Mask model, which was trained for 600 epochs with a larger batch size of 10,000). The hyperparameter λ in the MMD Representation and MMD Hybrid model was set to 1. The final model in all methods was the one that gave the best validation loss.

The results are summarized in Table 4. The performance reported is the average test accuracy from 10 runs for each model. Note that this baseline model can achieve around 98% test accuracy on the

original test dataset, but only around 87% accuracy on the corrupted test dataset. However, MMD Representation, MMD Mask, and MMD Hybrid can all improve the performance by a large margin, to 95.9%, 97.2%, and 97.2% accuracy, respectively. Because we corrupted the test set by applying patterned masks, MMD Mask is a more natural treatment compared to MMD Representation. Since MMD Hybrid combines MMD Representation and MMD Masks, it works as well as MMD Mask in this case. All three methods performed better than the two traditional data augmentation methods: these achieve accuracies of 93.4%and 95.0%. We believe that this is simply because the learned transformation is more effective and efficient compared to more random transformations. The golden model achieves 98.7% accuracy, which is not too far away from performances from the best MMD approaches. Figure 9 shows some samples of generated masks applied to the original training set, which are quite close to the true masks we imposed on the test set.

E Limitations and Future Work

With the current approaches introduced in this paper, the learned model is calibrated to a specific test set. If there is a new test set with a different distribution shift and/or missingness shift, we need to retrain the model with this new test set. We think that MMD has the potential to learn a more generalized model that is robust for different sets of test data if we combine this technique with meta-learning approaches. This is out of scope for this paper and an interesting area for future work to explore. However, in scenarios where one has a target test set to label, our proposed MMD approaches help the model learn to use the best features in both the input and embedding spaces and avoid dangerous extrapolation as much as possible, and as a result, achieve better performance on the test set.